MOILab: towards a labelled theorem prover for intuitionistic modal logics

Marianna Girlando, Marianela Morales

MOILab: towards a labelled theorem prover for intuitionistic modal logics

Marianna Girlando[1] and Marianela Morales[2]

[1] Inria, Equipe Partout, Centre Inria Saclay - Île-de-France & LIX, École Polytechnique, France
marianna.girlando@inria.fr

[2] LIX, École Polytechnique, France & Inria, Equipe Partout, Centre Inria Saclay - Île-de-France
marianela.morales@polytechnique.edu

Abstract

We present MOILab, a prototype Prolog theorem prover implementing a labelled sequent calculus for IK, the basic system in the intuitionistic modal logics family. MOILab builds upon MOIN, a theorem prover implementing nested sequent calculi (both single-conclusion and multi-conclusion) for all the logics in the modal intuitionistic cube. With respect to the nested implementations, MOILab offers a straightforward countermodel construction in case of proof search failure.

1 Introduction

We tackle the problem of defining automated theorem provers for intuitionistic modal logics. As the name says, intuitionistic modal logics are an intuitionistic version of (classical, normal) modal systems. We here consider the intuitionistic modal systems introduced in [3, 11] and studied in Simpson's Ph.D. thesis [12]. In analogy to what happens with modal logics, the basic system of intuitionistic modal logics (IK) can be extended with a set of axioms, generating 15 logics organised into the intuitionistic modal logic "cube" [13]. In this paper, however, we are concerned only with IK. Several proof systems for intuitionistic modal logics have been defined, among which, single-conclusion (or Gentzen-style) nested sequents, [13, 9, 2], multi-conclusion (or Maehara-style) nested sequents [5] and labelled calculi [8].

In [4] is presented a SWI Prolog theorem prover for classical and intuitionistic modal logics, called MOIN[1]. The prover implements nested proof systems: nested sequents from [1] for classical modal logics and, for the logics in the intuitionistic modal cube, it implements both single-conclusion nested sequents from [13] and multi-conclusion nested sequents from [5]. There are several other Prolog prover implementing nested sequents: refer to [7, 6] for a Prolog implementation of nested sequents for non-normal modal logics, and to [10] for normal conditional logics. MOIN implementation is slightly different, in that the data structure chosen to represent nested sequents is a list instead of a tree of lists. For the systems whose decidability is known, MOIN terminates[2].

We here present a prototype Prolog prover extending MOIN and implementing a labelled sequent calculus for IK, the basic system of intuitionistic modal logics. The prover is called MOILab, for *MOdal and Intuitionistic Labelled sequents*[3]. The labelled proof system, introduced in [8], internalises the semantic information from bi-relational models for intuitionistic modal logics into the sequent calculus syntax. As a result, the calculus is equipped with two relation

[1] MOIN stands for *MOdal and Intuitionistic Nested sequents*. The prover is available here: http://www.lix.polytechnique.fr/Labo/Lutz.Strassburger/Software/Moin/MoinProver.html

[2] For the record, all systems of intuitionistic modal logics are decidable, except for IK4, ID4 and IS4.

[3] MOILab is available here: http://mariannagirlando.com/MOILab.html

symbols, one for the accessibility relation from Kripke semantics for modal logics and one for the preorder relation from Kripke semantics for intuitionistic propositional logic.

With respect to the nested systems for intuitionistic modal logics, the labelled calculus offers two main advantages: since all its rules are invertible, no backtrack points need to be introduced in proof search, and a countermodel can be easily extracted from the upper sequent of a failed branch. This motivates the introduction of MOILab. As for now, the theorem prover is a prototype: only the basic logic IK is implemented, and the implemented proof search might not terminate on some sequents.

The paper is organised as follows: Section 2 introduces the syntax and semantics of intuitionistic modal logic IK, and Section 3 presents the main features of MOILab. For a presentation of the labelled sequent calculus, the reader is referred to [8].

2 Intuitionistic modal logic IK

The language of intuitionistic modal logics extends the language of intuitionistic propositional logic with the modal operators $\Box$ and $\Diamond$. Lacking the De Morgan duality, there are several variants of the *distributivity axiom* that are classically but not intuitionistically equivalent. An intuitionistic variant of modal logic K, called IK, is obtained by adding to an axiomatization of intuitionistic propositional logic the *necessitation rule* of K and the following axioms[4]:

$$\begin{array}{lll} \mathsf{k}_1: \Box(A \supset B) \supset (\Box A \supset \Box B) & \mathsf{k}_3: \Diamond(A \vee B) \supset (\Diamond A \vee \Diamond B) & \mathsf{k}_5: \Diamond\bot \supset \bot \\ \mathsf{k}_2: \Box(A \supset B) \supset (\Diamond A \supset \Diamond B) & \mathsf{k}_4: (\Diamond A \supset \Box B) \supset \Box(A \supset B) & \end{array}$$

Bi-relational models for IK [3, 11, 12] are defined by adding a valuation for atomic formulas to a bi-relational frame (refer to [8] for details):

Definition 2.1. A *bi-relational frame* $\mathcal{F}$ is a triple $\langle W, R, \leq \rangle$ of a set of worlds W equipped with an accessibility relation R and a preorder $\leq$ (*i.e.* a reflexive and transitive relation) satisfying:

(F_1) For all $u, v, v' \in W$, if uRv and $v \leq v'$, there exists u' s.t. $u \leq u'$ and $u'Rv'$.

$$\begin{tikzcd} u' \arrow[r, dotted, "R"] & v' \\ u \arrow[u, dotted, "\leq"] \arrow[r, "R"] & v \arrow[u, "\leq"'] \end{tikzcd}$$

(F_2) For all $u, u', v \in W$, if uRv and $u \leq u'$, there exists v' s.t. $u'Rv'$ and $v \leq v'$.

$$\begin{tikzcd} u' \arrow[r, dotted, "R"] & v' \\ u \arrow[u, "\leq"] \arrow[r, "R"] & v \arrow[u, dotted, "\leq"'] \end{tikzcd}$$

The accessibility relation R comes from Kripke frames for modal logics, and xRy is usually interpreted as "world y is accessible from world x". The preorder relation $\leq$ comes from Kripke frames for intuitionistic propositional logic, and can be interpreted as expressing a time relation between worlds: $x \leq y$ means "world y is a future of world x".

Reflecting the definition of bi-relational models, the sequents of the labelled calculus $\mathsf{labIK}_{\leq}$ defined in [8] are equipped with two relation symbols, one for R and one for $\leq$.

[4] We employ the coloured syntax from [8]: variables for labels are blue and formulas are green. The aim is to improve readability.

Definition 2.2. A two-sided intuitionistic *labelled sequent* is of the form $\mathcal{R}, \Gamma \Longrightarrow \Delta$ where $\mathcal{R}$ denotes a set of relational atoms xRy and preorder atoms $x \leq y$, and Γ and Δ are multi-sets of labelled formulas $x{:}A$ (for x and y variables for labels and A intuitionistic modal formula).

The rules of the labelled calculus for IK, called $\mathsf{labIK}_{\leq}$, can be found in [8]. Moreover, in [8] it is proved that all $\mathsf{labIK}_{\leq}$ rules are invertible (Lemma 6.4), and that the cut rule is admissible (Theorem 6.1). The rules needed to extend $\mathsf{labIK}_{\leq}$ to logics whose axiomatization extends IK by *one-sided intuitionistic Scott-Lemmon axioms*, i.e., axioms of the form $\Box^k\Diamond^l A \supset \Box^m\Diamond^n A$, for k, l, m, n natural numbers, are also defined.

Termination of proof search with $\mathsf{labIK}_{\leq}$ is not proved in [8]. As it is often the case with labelled proof systems, proving termination presents some difficulties: since all the rules are invertible, the sequent grows when going from the conclusion to the premiss(es) of each rule, and one cannot check for repetition of whole sequents in a proof search branch. Moreover, while the unlabelled formulas[5] occurring in proof search are finitely many, and all are subformulas of the (unlabelled) formula at the root, these formulas could be labelled with infinite ever-changing labels, thus giving rise to infinite branches. As a consequence, completeness of the calculus is established in [8] by means of cut-admissibility, and not by means of a countermodel construction from failed proof search.

3 Towards a labelled theorem prover

MOILab implements the labelled sequent calculus $\mathsf{labIK}_{\leq}$ from [8]. The prover is composed of a set of clauses, each implementing a rule of the labelled sequent calculus. The only exception is the rule of reflexivity, which does not have a dedicated clause and is instead applied together with the rules introducing (backwards) a new label.

Overall, MOILab builds upon the structure of MOIN: labelled sequents are represented by means of Prolog lists, in which each element is a pair comprising a label (an integer) and a formula. Separate lists store the accessibility relations and the preorder relations among labels.

Propositional variables are represented in MOILab syntax as Prolog atoms `a`,`b`,...; $\bot$ and $\top$ are Prolog `false` and `true`, and the connectives $\neg$, $\wedge$, $\vee$, $\supset$, $\Box$ and $\Diamond$ are respectively represented by `~`, `v`, `^`, `->`, `!` and `?`. Labelled sequents are represented by means of two Prolog lists `Fut,Rel,Gamma,Delta`. `Gamma` and `Delta` are lists of triples `(X,F,Sign)`, where `F` is a formula in MOILab syntax, `X` is the label of `F`, i.e., an integer, and `Sign` is either `+` or `-`. Rules can only be applied to formulas with a positive sign, while formulas with a negative sign are used for book-keeping. `Fut` and `Rel` are lists of pairs `(X,Y)`, respectively representing the preorder relation and the accessibility relation between labels.

Proof search is invoked by the predicate `derive(F)`, where `F` is the formula to be checked. For instance, `derive(((?a)-> (!b)) -> (!(a->b)))` triggers the derivation of axiom k_4 in $\mathsf{labIK}_{\leq}$. The predicate `derive` queries the predicate `prove_lab\4`, responsible of the actual proof search. The predicate is recursively invoked and generates the proof-search tree for the formula. The application of `prove` to a branch stops when an axiom clause is reached (success), or when no clause succeeds, producing a failed branch. However, since a full termination strategy is missing, it might happen that proof search never stops. If proof search stops and produces a success, MOILab gives in output a LaTeX file containing the derivation. If proof search stops producing a failure, MOILab prints out a countermodel in a LaTeX file.

[5] For unlabelled formula we mean a labelled formula in which we ignore the label: thus, the unlabelled formula corresponding to $x{:}A$ is A.

Termination is still an issue: as for now, MOILab implements the naive strategy of not applying a rule to a labelled formula if the labelled formula to be introduced already occurs in the sequent. This is not enough to ensure termination of proof search, and it might be case that, on some formulas, proof search goes on forever.

Strategies ensuring termination of proof search do exists for nested calculi (both single- and multi-conclusion) for IK: in fact, proof search in MOIN, the prover implementing these nested proof systems, terminates for IK. The termination strategy for nested sequents basically checks for repetition of sequents in a derivation branch (refer to [4] for details on the termination strategy implemented in MOIN). However, this strategy cannot be directly applied to the labelled calculus, where each formula has a label: it might happen that the same formula is labelled by infinite different labels. Moreover, since all rules of the labelled calculus are invertible, the sequent always grows when applying bottom-up rules of the calculus. Thus, the check for repetition needs to be performed within the same sequent, and taking the labels of formulas into account. A more refined termination strategy is currently under study.

Thanks to invertibility of the $\mathsf{labIK}_{\leq}$ rules, backtrack points do not need to be introduced in proof search. The countermodel extraction is straightforward, since no information is lost in going from the conclusion to the premiss(es) of the rules, and only the upper sequent of a failed derivation branch needs to be considered. With nested calculi, not all the rules are invertible, and the process of countermodel construction from failed proof search requires some more work: other than the upper sequent of a failed branch, one needs to take into account all the sequents in the branch occurring as conclusion of non-invertible rules.

4 Conclusions

The most important missing feature of MOILab is a termination of proof search, which is object of current study. Our immediate goal is to define and implement a termination strategy for $\mathsf{labIK}_{\leq}$ which is general enough to be applied, modulo some modification, to labelled calculi for extensions of IK.

Our long term goal is to define a theorem prover modularly implementing labelled calculi for logics extending IK. These systems comprise both the extensions of IK by means of one-sided intuitionistic Scott-Lemmon axioms, whose labelled rules are defined in [8], and extensions of IK with intuitionistic variants of modal axioms d, t, b, 4 and 5 [13], which correspond to the logics in the intuitionistic modal cube.

References

[1] Kai Brünnler. Deep sequent systems for modal logic. *Archive for Mathematical Logic*, 48(6):551–577, 2009.

[2] Kaustuv Chaudhuri, Sonia Marin, and Lutz Straßburger. Modular focused proof systems for intuitionistic modal logics. In *1st International Conference on Formal Structures for Computation and Deduction (FSCD 2016)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

[3] Giselle Fischer-Servi. Axiomatizations for some intuitionistic modal logics. *Rend. Sem. Mat. Univers. Politecn. Torino*, 42(3):179–194, 1984.

[4] Marianna Girlando and Lutz Straßburger. Moin: A nested sequent theorem prover for intuitionistic modal logics (system description). *IJCAR 2020*, forthcoming.

[5] Roman Kuznets and Lutz Straßburger. Maehara-style modal nested calculi. *Archive for Mathematical Logic*, 58(3-4):359–385, 2019.

[6] Björn Lellmann. Combining monotone and normal modal logic in nested sequents–with countermodels. In *International Conference on Automated Reasoning with Analytic Tableaux and Related Methods*, pages 203–220. Springer, 2019.

[7] Björn Lellmann and Elaine Pimentel. Modularisation of sequent calculi for normal and non-normal modalities. *ACM Transactions on Computational Logic (TOCL)*, 20(2):7, 2019.

[8] Sonia Marin, Marianela Morales, and Lutz Straßburger. A fully labelled proof system for intuitionistic modal logics. Working paper or preprint, hal-02390454, September 2019.

[9] Sonia Marin and Lutz Straßburger. Label-free Modular Systems for Classical and Intuitionistic Modal Logics. In *Advances in Modal Logic 10*, Groningen, Netherlands, August 2014.

[10] Nicola Olivetti and Gian Luca Pozzato. Nescond: an implementation of nested sequent calculi for conditional logics. In C. Weidenbach S. Demri, D. Kapur, editor, *International Joint Conference on Automated Reasoning*, pages 511–518. Springer, 2014.

[11] Gordon Plotkin and Colin Stirling. A framework for intuitionistic modal logics. In *Proceedings of the 1986 Conference on Theoretical aspects of reasoning about knowledge*, pages 399–406. Morgan Kaufmann Publishers Inc., 1986.

[12] Alex Simpson. *The Proof Theory and Semantics of Intuitionistic Modal Logic*. PhD thesis, University of Edinburgh, 1994.

[13] Lutz Straßburger. Cut elimination in nested sequents for intuitionistic modal logics. In Frank Pfenning, editor, *FoSSaCS'13*, volume 7794 of *LNCS*, pages 209–224. Springer, 2013.